

IN THE CLAIMS

This listing of the claims will replace all prior versions and listings of claim in the present application.

RECEIVED
CENTRAL FAX CENTER

Listing of Claims

JAN 29 2007

1. (currently amended) A method of detecting invalid memory access used in a computer which executes a language system having a specific memory management function; a first program code that is executed under the control of the language system, and that accesses a first memory area reserved by the language system; and a second program code that is directly executed under the control of OS, and that accesses a second memory area reserved by the OS, wherein the first program code and the second program code have such an interrelationship that the first program code is a calling program code while the second program code is a program code invoked by the first program code, and wherein said method executed by the language system detects invalid memory access to the first memory area caused by the second program code, said method executed by the language system comprising the steps of:
 - | allowing said language system to set the memory protection of the first memory area before the first program code calls the second program code;
 - | calling and executing the second program code from the first program code;
 - | when a memory protection exception occurs, notifying of invalid memory access caused by the second program code to outside; and
 - | when the execution of the second program code ends and the control returns to the language system, disabling the memory protection of the first memory area.

2. (original) A method of detecting invalid memory access according to Claim 1, wherein:

when said memory protection exception occurs, if it is detected that the first program code performs normal memory access to the first memory area, said language system disables the memory protection to allow the normal memory access, and then enables the memory protection again.

3. (original) A method of detecting invalid memory access according to Claim 1, wherein:

if the first program code is executed under the multithread control, said language system suspends the execution of other threads while a certain thread calls the second program code.

4. (currently amended) A method of detecting invalid memory access used in a computer which executes a language system having a specific memory management function; a first program code that is executed under the control of the language system, and that accesses a first memory area reserved by the language system; and a second program code that is directly executed under the control of OS, and that accesses a second memory area reserved by the OS; wherein said method executed by the language system detects invalid memory access to the first memory area caused by the second program code, said method executed by the language system comprising the steps of:

~~allowing said language system to save~~saving code information associated with the contents of the first memory area before the first program code calls the second program code;

calling and executing the second program code from the first program code;
when the execution of the second program code ends and the control returns
to the language system, judging whether or not code information associated with the
contents of the first memory area coincides with the saved code information; and
if the code information associated with the contents of the first memory area
does not coincide with the saved code information, notifying of invalid memory
access caused by the second program code to outside,
wherein, when it is detected that while the second program code is called the
first program code normally updates the first memory area, said language system
updates the saved code information based on code information associated with
contents of the first memory area updated,

Claim 5 (canceled).

6. (original) A method of detecting invalid memory access according
to Claim 4, wherein:
if the first program code is executed under the multithread control, said
language system suspends the execution of other threads while a certain thread
calls the second program code.

7. (currently amended) A ~~program used in a computer which~~
executes computer readable medium having a computer program recorded thereon
for execution on a computer for realizing a language system having a specific
memory management function; a first program code that is executed under the
control of the language system, and that accesses a first memory area reserved by

the language system; and a second program code that is directly executed under the control of OS, and that accesses a second memory area reserved by the OS; wherein the first program code and the second program code have such an interrelationship that the first program code is a calling program code while the second program code is a calling program code invoked by the first program code, and said program allowing said computer to execute language system's functions of detecting the language system detects invalid memory access to the first memory area caused by the second program code; the computer program causing the computer to perform the following method steps:

— wherein said computer executes the functions of:

- setting the memory protection of the first memory area before the first program code calls the second program code;
- calling and executing the second program code from the first program code;
- when a memory protection exception occurs, notifying of invalid memory access caused by the second program code to outside; and
- when the execution of the second program code ends and the control returns to the language system, disabling the memory protection of the first memory area.

8. (currently amended) A program computer readable medium

according to Claim 7, allowing the computer to execute the functions the computer program further causing the computer to perform the method step of:

when said memory protection exception occurs, if it is detected that the first program code performs normal memory access to the first memory area, disabling the memory protection, allowing the normal memory access, and enabling the memory protection again.

9. (currently amended) A program-computer readable medium
according to Claim 7, ~~allowing the computer to execute the function the computer~~
~~program further causing the computer to perform the method step of:~~

If the first program code is executed under the multithread control, suspending
the execution of other threads while a certain thread calls the second program code.

10. (currently amended) A program used in a computer which
executes computer readable medium having a computer program recorded thereon
for execution on a computer for realizing a language system having a specific
memory management function; a first program code that is executed under the
control of the language system, and that accesses a first memory area reserved by
the language system; and a second program code that is directly executed under the
control of OS, and that accesses a second memory area reserved by the OS; ~~said~~
~~program allowing said computer to execute language system's functions of detecting~~
~~wherein the language system detects~~ invalid memory access to the first memory
area caused by the second program code; the computer program causing the
computer to perform the following method steps of:

~~— wherein said computer executes the functions of:~~
saving code information associated with the contents of the first memory area
before the first program code calls the second program code;
calling and executing the second program code from the first program code;
when the execution of the second program code ends and the control returns
to the language system, judging whether or not code information associated with the
contents of the first memory area coincides with the saved code information; and

if the code information associated with the contents of the first memory area does not coincide with the saved code information, notifying of invalid memory access caused by the second program code to outside.

wherein, when it is detected that while the second program code is called the first program code normally updates the first memory area, said language system updates the saved code information based on code information associated with contents of the first memory area updated.

Claim 11 (canceled).

12. (currently amended) A program-computer readable medium according to Claim 10, ~~allowing the computer to function the computer program further causing the computer to perform the method step of:~~

if the first program code is executed under the multithread control, suspending the execution of other threads while a certain thread calls the second program code.

13. (currently amended) A language system used in a computer which executes a language system having a specific memory management function; a first program code that is executed under the control of the language system, and that accesses a first memory area reserved by the language system; and a second program code that is directly executed under the control of OS, and that accesses a second memory area reserved by the OS; wherein the first program code and the second program code have such an interrelationship that the first program code is a program code invoked by the first program code, and wherein said language system

detects invalid memory access to the first memory area caused by the second program code, said language system comprising:

means for setting memory protection of the first memory area before the first program code calls the second program code, for calling and executing the second program code from the first program code, and for notifying of invalid memory access caused by the second program code to outside when a memory protection exception occurs; and

means for disabling the memory protection when the execution of the second program code ends and the control returns to the language system.

14. (original) A language system according to Claim 13, further comprising:

means, when said memory protection exception occurs, if it is detected that the first program code performs normal memory access to the first memory area, for disabling the memory protection, allowing the normal memory access, and then enabling the memory protection again.

15. (original) A language system according to Claim 13, further comprising:

means, if the first program code is executed under the multithread control, for suspending the execution of the other threads while a certain thread calls the second program code.